EDS_230_Growth_Assignment

Clarissa Boyajian, Ryan Munnikhuis, and Marie Rivers

2022-05-15

Growth Assignment

```
library(deSolve)
library(here)
```

```
## here() starts at /Users/marierivers/Documents/UCSB_Environmental_Data_Science/EDS_230_Modeling_Envir
```

```
library(sensitivity)
```

```
## Registered S3 method overwritten by 'sensitivity':
##   method    from
##   print.src dplyr
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.1.2

## Warning: package 'tibble' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract() masks sensitivity::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x dplyr::src()     masks sensitivity::src()
```

```r
library(patchwork)
```

Background

Consider the following model of forest growth (where forest size is measured in units of carbon (C)):

$$dC/dt = r * C$$

for forest where **C is below a threshold canopy closure** and

$$dC/dt = g * (1 = C/K)$$

for forests where **carbon is at or above the threshold canopy closure**

and K is a carrying capacity in units of Carbon. The size of the forest (C), canopy closure threshold and carrying capacity are all in units of carbon. Think of the canopy closure threshold as the size of the forest at which growth rates change from exponential to linear. Think of r as the early exponential growth rate and g as the linear growth rate once canopy closure has been reached.

C = forest size, units of carbon

T = time, years

r = early exponential growth rate (before canopy closure has been reached)

g = the linear growth rate once canopy closure has been reached

K = carrying capacity, units of carbon

canopy closure threshold = the size of the forest at which growth rates change from exponential to linear

1. Implement this model in R as a differential equation

```r
source(here("R", "dforest_growth.R"))

dforest_growth
```

```
## function (Time, C, parms)
## {
##     if (C < parms$canopy_threshold) {
##         dC_dt <- parms$r * C
##     }
##     else if (C >= parms$canopy_threshold) {
##         dC_dt <- parms$g * (1 - C/parms$K)
##     }
##     return(list(dC_dt))
## }
```

2. Run the model for 300 years (using the ODE solver) starting with an initial forest size of 10 kg Carbon and using the following parameters: canopy closure threshold = 50 kg Carbon

K = 250 kg C (carrying capacity)

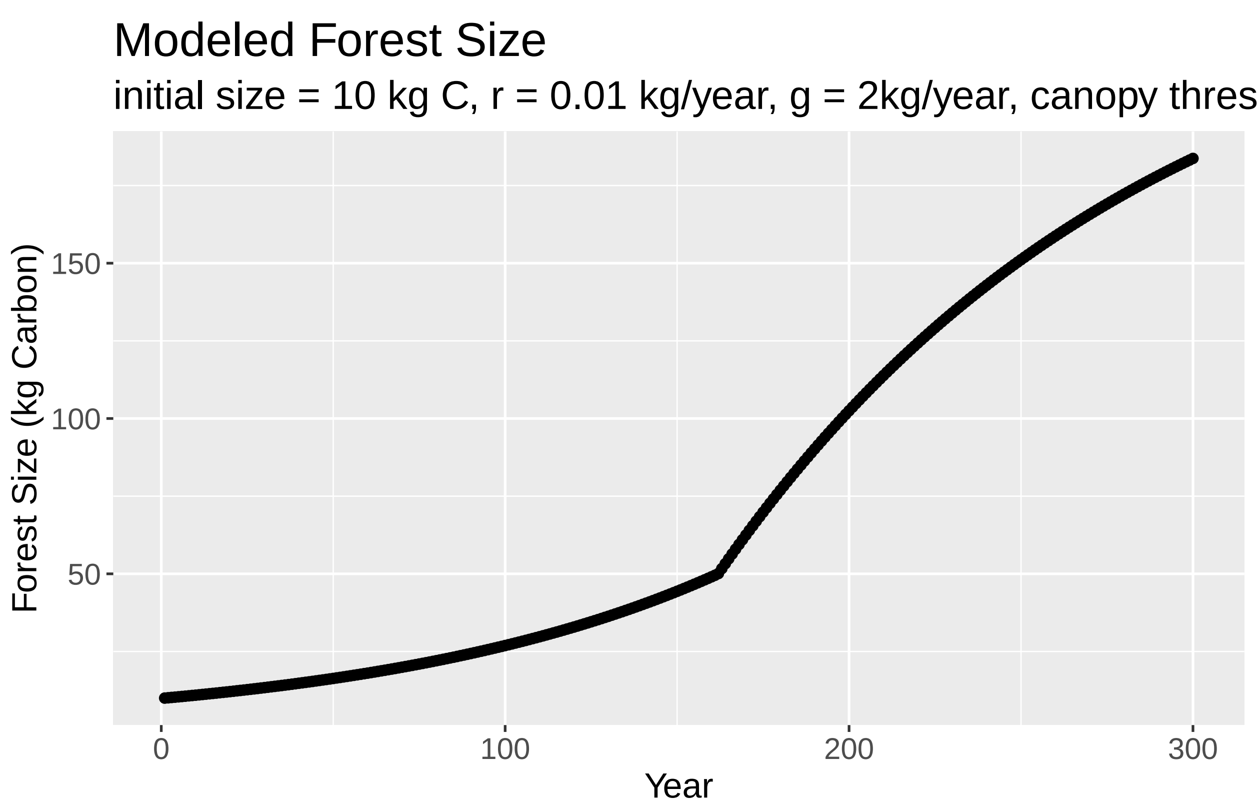
r = 0.01 (exponential growth rate before canopy closure)

g = 2 kg/year (linear growth rate after canopy closure)

```r
years <- seq(from = 1, to = 300) # time sequence of years
Cinitial <- 10 # initial forest size, kg Carbon
canopy_threshold <- 50 # kg Carbon
K <- 250 # carrying capacity, kg C
r <- 0.01 # exponential growth rate before canopy closure)
g <- 2 # linear growth rate after canopy closure, kg/year

# set up parameter list
parms <- list(canopy_threshold = canopy_threshold, K = K, r = r, g = g)

results <- ode(y = Cinitial, times = years, func = dforest_growth, parms = parms, method = "lsoda")

colnames(results) = c("year", "carbon")
```

Graph the results

```r
carbon_graph <- ggplot(data = as.data.frame(results),
                       aes(x = year, y = carbon)) +
  geom_point() +
  labs(x = "Year", y = "Forest Size (kg Carbon)",
       title = "Modeled Forest Size",
       subtitle = paste0("initial size = ", Cinitial, " kg C, r = ",
                         r, " kg/year, g = ", g, "kg/year, canopy threshold = ",
                         canopy_threshold, "kg C, carrying capacity = ", K, " kg C")) +
  theme(title = element_text(size = 16),
        axis.title = element_text(size = 14),
        axis.text = element_text(size = 12))
carbon_graph
```



```
# check that units are stated correctly
```

3. Run a sobol sensitivity analysis that explores how the estimated maximum and mean forest size (e.g. maximum and mean values of C over the 300 years) varies with the pre canopy closure growth rate (r) and post-canopy closure rate (g) and canopy closure threshold and carrying capacity (K).

Assume that the parameters are all normally distributed with means as given above and standard deviation of 10% of mean value.

```
# want to learn about sensitivity with all the parameters normally distributed
# set the number of parameters
np <- 100

r <- rnorm(mean = r, sd = 0.1 * r, n = np)
g <- rnorm(mean = g, sd = 0.1 * g, n = np)
canopy_threshold <- rnorm(mean = canopy_threshold,
                          sd = 0.1 * canopy_threshold, n = np)
K <- rnorm(mean = K, sd = 0.1 * K, n = np)
X1 = cbind.data.frame(r = r, g = g, canopy_threshold = canopy_threshold, K = K)

# repeat to get our second set of samples
r <- rnorm(mean = r, sd = 0.1 * r, n = np)
g <- rnorm(mean = g, sd = 0.1 * g, n = np)
canopy_threshold <- rnorm(mean = canopy_threshold, sd = 0.1 * canopy_threshold, n = np)
K <- rnorm(mean = K, sd = 0.1 * K, n = np)
X2 = cbind.data.frame(r = r, g = g, canopy_threshold = canopy_threshold, K = K)

# create  sobel object and get sets of parameters for running the model
sens_Carbon = sobolSalt(model = NULL, X1, X2, nboot = 300)

# our parameter sets are
head(sens_Carbon$X)
```

```
##             [,1]     [,2]     [,3]     [,4]
## [1,] 0.009294896 2.121923 46.91205 243.0616
## [2,] 0.009268179 2.200607 48.54521 271.3496
## [3,] 0.009817506 1.978199 55.03924 234.3816
## [4,] 0.008562605 2.206238 50.86886 316.7523
## [5,] 0.011972388 2.038825 48.77114 245.1581
## [6,] 0.010929025 1.996804 54.16936 244.0424
```

```
# lets add names
colnames(sens_Carbon$X) = c("r", "g", "canopy_threshold", "K")
head(sens_Carbon$X)
```

```
##                r        g canopy_threshold        K
## [1,] 0.009294896 2.121923         46.91205 243.0616
## [2,] 0.009268179 2.200607         48.54521 271.3496
## [3,] 0.009817506 1.978199         55.03924 234.3816
## [4,] 0.008562605 2.206238         50.86886 316.7523
## [5,] 0.011972388 2.038825         48.77114 245.1581
## [6,] 0.010929025 1.996804         54.16936 244.0424
```

Create two additional functions that will compute the metrics we want and run the ODE solver and compute the metrics (a wrapper function that is a workflow to call the ODE solver and then compute metrics)

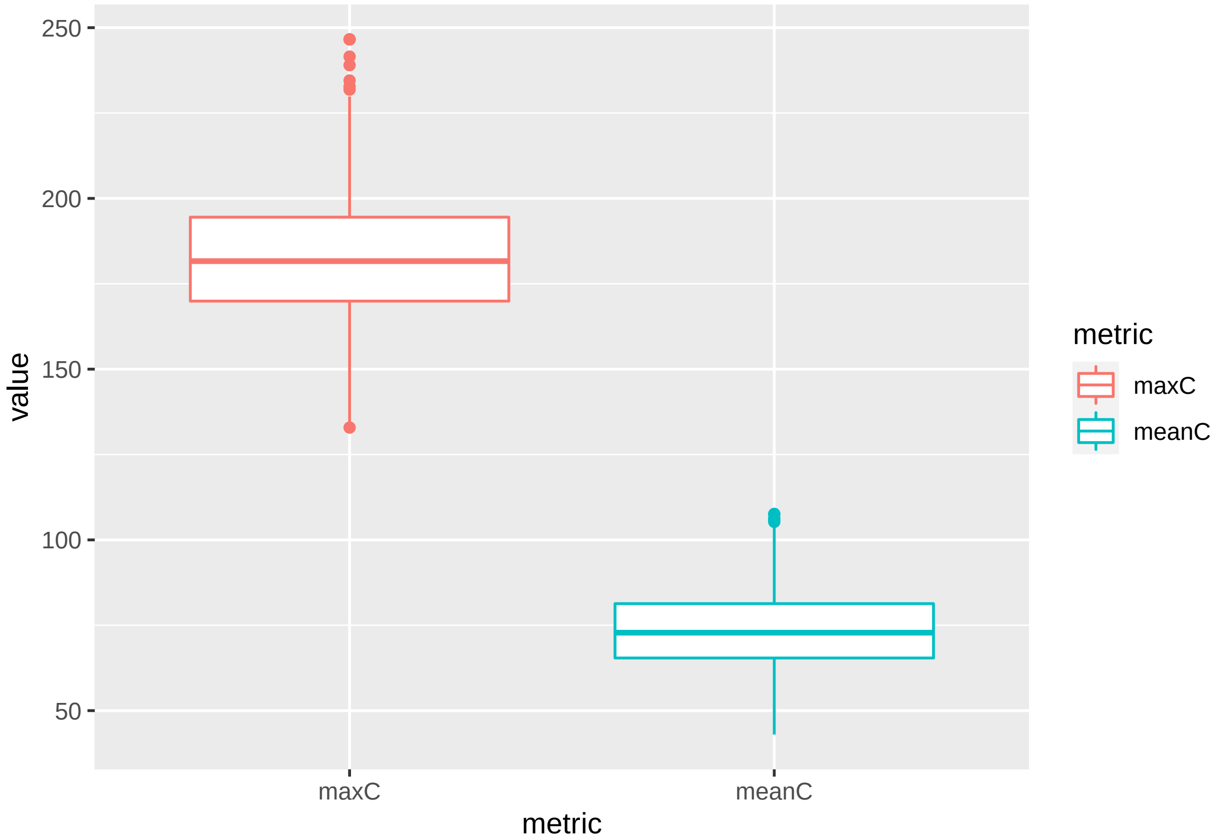
```
compute_metrics <- function(result) {

  maxC <- max(result$carbon)
  meanC <- mean(result$carbon)
  return(list(maxC = maxC, meanC = meanC))

}

# try function on first parameter set
compute_metrics(as.data.frame(results))
```

```
## $maxC
## [1] 183.7213
##
## $meanC
## [1] 72.98569
```

```
# need to apply the ode and the compute metrics function to all parameters
# define a wrapper function to do everything we need (run solver and compute metrics) and send back res

p_wrapper <- function(r, g, canopy_threshold, K, Cinitial, simtimes, func) {

  parms <- list(r = r, g = g, canopy_threshold = canopy_threshold, K = K)
  results <- ode(y = Cinitial, times = years, func = func, parms = parms, method = "lsoda")
  colnames(results) <- c("year", "carbon")

  # get metrics
  metrics <- compute_metrics(as.data.frame(results))

  return(metrics)

}

# # now use pmap
# allresults <- as.data.frame(sens_Carbon$X) %>%
#   pmap(p_wrapper, Cinitial = Cinitial, simtimes = years, func = dforest_growth)
#
# # extract out results from pmap into a data frame
# allres <- allresults %>%
#   map_dfr(`[`, c("maxC", "meanC"))

## Clarissa note: can shorten process above by using `pmap_dfr` which returns results as an rbound data
allres <- as.data.frame(sens_Carbon$X) %>%
  pmap_dfr(p_wrapper, Cinitial = Cinitial, simtimes = years, func = dforest_growth)
```

Graph the results of the sensitivity analysis as a box plot of maximum forest size and a plot of the two Sobol indices (S and T)

```r
# create boxplot
tmp <- allres %>%
  pivot_longer(cols = 1:2,
               names_to = "metric",
               values_to = "value")

ggplot(data = tmp,
       aes(x = metric, y = value, col = metric)) +
  geom_boxplot()
```




```r
sens_Carbon_maxC <- sensitivity::tell(sens_Carbon, allres$maxC)
```

```r
# S (first order index)
maxC_S <- sens_Carbon_maxC$S %>%
  rowid_to_column("parameter") %>%
  mutate(parameter = str_replace(string = parameter, pattern = "1", replacement = "r")) %>%
  mutate(parameter = str_replace(string = parameter, pattern = "2", replacement = "g")) %>%
  mutate(parameter = str_replace(string = parameter, pattern = "3", replacement = "canopy_threshold")) %
  mutate(parameter = str_replace(string = parameter, pattern = "4", replacement = "K"))
```

```r
# T (total sensitivity index)
maxC_T <- as.data.frame(sens_Carbon_maxC$T) %>%
  rowid_to_column("parameter") %>%
  mutate(parameter = str_replace(string = parameter, pattern = "1", replacement = "r")) %>%
  mutate(parameter = str_replace(string = parameter, pattern = "2", replacement = "g")) %>%
```

```r
  mutate(parameter = str_replace(string = parameter, pattern = "3", replacement = "canopy_threshold")) %
  mutate(parameter = str_replace(string = parameter, pattern = "4", replacement = "K"))
maxC_T
```

```
##          parameter   original         bias  std. error   min. c.i.  max. c.i.
## 1                r 0.07834384 0.0042083939 0.020631057 0.021451794 0.10809310
## 2                g 0.06262575 0.0024044985 0.014931206 0.023669651 0.08481200
## 3 canopy_threshold 0.02189415 0.0006643479 0.005575418 0.008412523 0.03034692
## 4                K 0.14148630 0.0017167289 0.028781659 0.076176846 0.18365560
```

```r
# X1 = r
# X2 = g
# X3 = canopy_threshold
# X4 = K
```

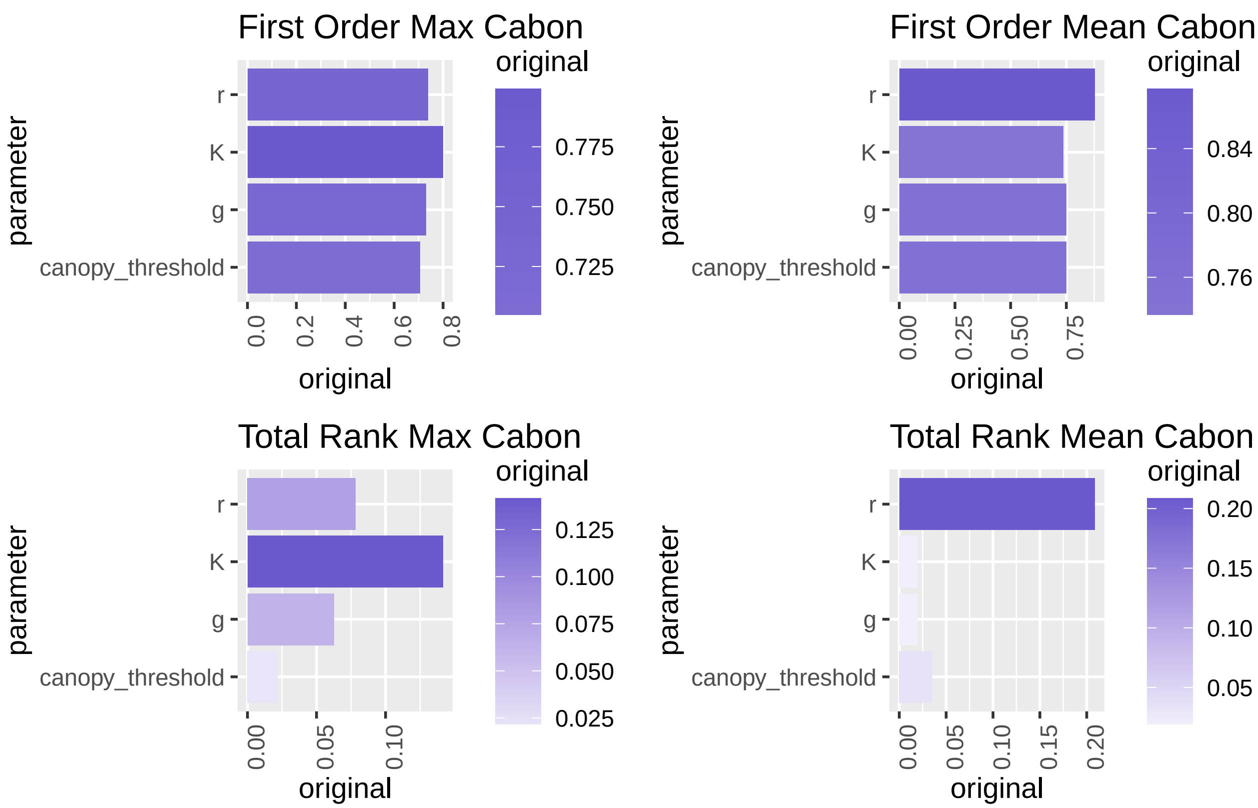
```r
sens_Carbon_meanC <- sensitivity::tell(sens_Carbon, allres$meanC)
```

```r
# S (first order index)
meanC_S <- sens_Carbon_meanC$S %>%
  rowid_to_column("parameter") %>%
  mutate(parameter = str_replace(string = parameter, pattern = "1", replacement = "r")) %>%
  mutate(parameter = str_replace(string = parameter, pattern = "2", replacement = "g")) %>%
  mutate(parameter = str_replace(string = parameter, pattern = "3", replacement = "canopy_threshold")) %
  mutate(parameter = str_replace(string = parameter, pattern = "4", replacement = "K"))
```

```r
# T (total sensitivity index)
meanC_T <- sens_Carbon_meanC$T %>%
  rowid_to_column("parameter") %>%
  mutate(parameter = str_replace(string = parameter, pattern = "1", replacement = "r")) %>%
  mutate(parameter = str_replace(string = parameter, pattern = "2", replacement = "g")) %>%
  mutate(parameter = str_replace(string = parameter, pattern = "3", replacement = "canopy_threshold")) %
  mutate(parameter = str_replace(string = parameter, pattern = "4", replacement = "K"))
```

```r
maxC_S_plot <- ggplot(maxC_S) +
  geom_col(aes(y=original, x=parameter, fill=original),position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_gradient2(low='firebrick', high='slateblue') + ggtitle("First Order Max Cabon") +
  coord_flip()
# based on this, canopy threshold is the least important for max carbon and r is the most important

maxC_T_plot <- ggplot(maxC_T) +
  geom_col(aes(y=original, x=parameter, fill=original),position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_gradient2(low='firebrick', high='slateblue') + ggtitle("Total Rank Max Cabon") +
  coord_flip()

meanC_S_plot <- ggplot(meanC_S) +
  geom_col(aes(y=original, x=parameter, fill=original),position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_gradient2(low='firebrick', high='slateblue') + ggtitle("First Order Mean Cabon") +
  coord_flip()
```

```r
meanC_T_plot <- ggplot(meanC_T) +
  geom_col(aes(y=original, x=parameter, fill=original),position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_gradient2(low='firebrick', high='slateblue') + ggtitle("Total Rank Mean Cabon") +
  coord_flip()

S_T_plots <- (maxC_S_plot + meanC_S_plot) / (maxC_T_plot + meanC_T_plot)
S_T_plots
```



Discuss what the results of the simulation might mean for climate change impacts on forest growth (e.g. think about what parameters climate change might influence).

The parameter `r`, which is the exponential growth rate used before canopy closure has been reached, contributes most to both maximum carbon and mean carbon when considered on its own and when interacting with other parameters. Climate change may affect the growth rate of young trees (ie trees that have not grown to canopy threshold) more than mature tree. Young trees are more susceptible to die if they experience a water shortage or are damaged by severe weather events because their root system and branch systems are not fully formed.